Integer Programming Formulation

1 Integer Programming Introduction

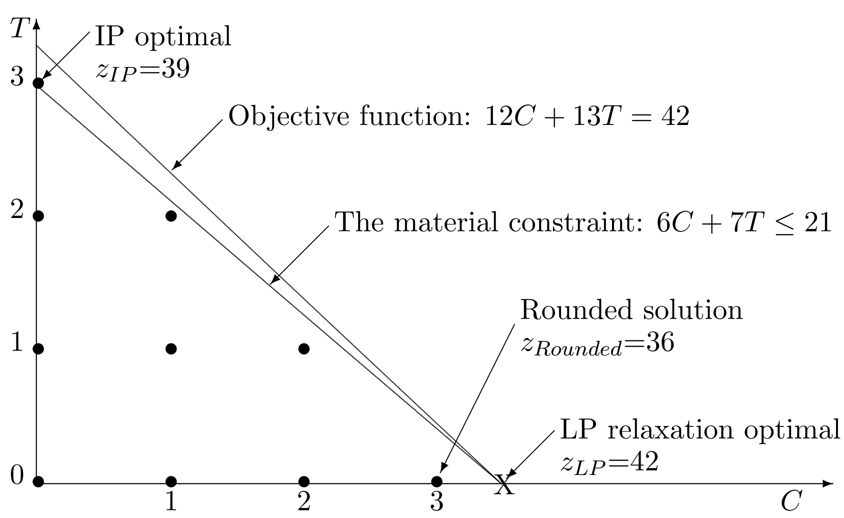
When we introduced linear programs in Chapter 1, we mentioned divisibility as one of the LP assumptions. Divisibility allowed us to consider activities in fractions: We could produce 7.8 units of a product, buy 12500.33 liters of oil, hire 12.123 people for full time, etc. Divisibility assumption is very defensible at times but not always. We can easily buy 12500.33 liters of oil but can not employ 12.123 people. Clearly some activities cannot be done in fractions and must be specified in integers for implementation. As soon as some of the activities are set to be integers, we are in Integer Programming domain. Formally, in an integer program some decision variables are forced to be integers.

We will give a small example here. Suppose we consider producing chairs and tables using only 21 m^2 of wood. Each chair (table) requires 6 (7) m^2 of wood. Each chair is sold at \$12 ($\times$10) and each table is sold at \$13 ($\times$10). Let C and T denote the number of tables and chairs produced. The IP formulation below maximizes the revenue:

$$
\begin{array}{llllll}
\text{Maximize :} & 12C & +13T & & & \\
\text{Subject to} & & & & & \\
& 6C & +7T & \leq & 21 & (1) \\
& C & ,\ T & \geq & 0 & (2) \\
& C & ,\ T & int & & (3)
\end{array}
$$

Note that this formulation differs from the corresponding LP formulation through constraint (3), which says that both C and T are integers.

For a practically oriented mind, solving an IP can be as straightforward as solving the associated LP and rounding the solution. To understand what can wrong with this approach, we will first solve the IP removing constraint (3) and round down (why not to round up?) the optimal values of C and T to satisfy (3). When the integer constraints are removed from an IP formulation, we obtain an LP formulation. This LP formulation is called the *LP relaxation*. Since we have only two decision variables, we will solve the LP relaxation graphically:



LP solution is (7/2,0) and is not integer so we round it down to (3,0). The objective value at (3,0) is

36. The optimal solution to IP is (0,3) with the objective value 39. 3 units of difference between objective value of the IP optimal and the rounded solution can be significantly higher in more complex problems. As a summary we cannot use rounded solutions of LP relaxations.

The toy example above has illustrated that solving IP's are not straightforward. For toy problems one can evaluate all the integer solutions in the feasible region and pick the best. However, for real problems this approach will take practically infinite amount of time. The solution procedures for IP's are still under development. Two approaches are common: *Branch and Bound technique*, and *Cutting planes.* These techniques are outside the scope of our discussion. Thus, we turn to integer programming formulations.

2 Knapsack Problem

Suppose that Jean Luc (an MBA student) plans to study 40 hours in a week. There are 8 courses he is considering to take in the spring term. They are listed below with the number of hours (per week) required to successfully complete each course.

Operations Research	Accout.	Info. Tech.	Finance	Market.	O. Behav.	Italian Cinema	Russian
9	7	5	8	5	3	7	10

Completing each of these courses increases Jean Luc's chances of finding a job at McKinsey's new Bucharest office. But the contributions of courses towards this dream are different as given below:

Operations Research	Accout.	Info. Tech.	Finance	Market.	O. Behav.	Italian Cinema	Russian
0.10	0.04	0.06	0.12	0.08	0.03	0.04	0.05

Jean Luc wants to choose his spring courses so that he maximizes his chances of getting this job. We will now formulate this situation as an IP. Before going any further, we assume that choice of one course does not affect the choice of any other, i.e., courses can be chosen independently.

2.1 Decision Variables

Jean Luc wants to know the set of the courses he should take. For each course, one by one, if Jean Luc can answer the question whether a course is in the set of courses to be taken, then he will know what he is taking. For example, if OR is in the set, the answer will be a "yes" for OR, otherwise a "no". Let us denote "yes" answers with 1 and "no" answers with 0. Also let x_{OR} be the question. Then, $x_{OR} = 1$ implies a "yes" answer for OR, it is taken. Conversely, $x_{OR} = 0$ implies that it is not taken. With the intuition developed here, we let

$$x_j = \left\{ \begin{array}{ll} 1 & \text{if course } j \text{ is taken} \\ 0 & \text{otherwise} \end{array} \right\}.$$

2.2 Constraints

There is only one constraint: 40 hours are available for studies. If OR is taken, it requires 9 hours per week. Otherwise, it requires 0 hours. In general, OR requires $9x_{OR}$ hours per week. Writing hourly requirements for each course and summing those up, we obtain the constraint:

$$9x_{OR} + 7x_{Ac} + 5x_{IT} + 8x_{Fi} + 5x_{Ma} + 3x_{OB} + 7x_{IC} + 10x_{Ru} \leq 40$$

2.3 Objective Function

When Jean Luc takes OR, he increases his chance of getting the job by 0.10. Otherwise his chance does not increase. In general, his chance increases by $0.10x_{OR}$. Writing contributions of each course to his chance and summing, we obtain the objective function:

$$\text{Maximize } 0.10x_{OR} + 0.04x_{Ac} + 0.06x_{IT} + 0.12x_{Fi} + 0.08x_{Ma} + 0.03x_{OB} + 0.04x_{IC} + 0.05x_{Ru}$$

2.4 IP Formulation

We put the objective function and the constraints together to obtain the formulation below:

Maximize $0.10x_{OR} + 0.04x_{Ac} + 0.06x_{IT} + 0.12x_{Fi} + 0.08x_{Ma} + 0.03x_{OB} + 0.04x_{IC} + 0.05x_{Ru}$
Subject to

$$9x_{OR} + 7x_{Ac} + 5x_{IT} + 8x_{Fi} + 5x_{Ma} + 3x_{OB} + 7x_{IC} + 10x_{Ru} \leq 40$$
$$x_j \in \{0, 1\}$$

So if we have a solution where $x_{OR} = 1$, $x_{Ac} = 1$, $x_{IT} = 1$, $x_{Fi} = 1$, $x_{Ma} = 1$ and $x_{OB} = 1$, then Operations Research, Accounting, Information Technology, Finance, Marketing, Organizational Behavior courses are taken. Taking these courses Jean Luc spends 37 hours per week for his course work. He then increases his chances by 0.43.

In this example all decision variables are binary, i.e., they are either 1 or 0. Such IP's are sometimes called Binary Programs. This example actually first originated from a camper considering what to put (food, soaps, magazines, mosquito repellents, etc.) into his fixed capacity knapsack so that he will have a comfortable camping experience. That is where the name knapsack comes from.

3 Facility Location Problem

This problem is very similar to the transportation problem. There are still n retailers receiving materials from warehouses. Only difference is that we want to decide on warehouse locations as well as flows. There are m locations where we can open up warehouses. These locations are known but whether we open up a warehouse at location i ($1 \leq i \leq m$) is not known. To open up a warehouse we pay a cost of F_i. The rest of data is the same as transportation problem data: supplies are b_i, demand are d_j and unit flow costs are c_{ij}. We want to meet customer demand at minimum cost and will build an IP for that purpose.

3.1 Decision Variables

This problem is a generalization of the transportation problem so we still need to decide on flows from warehouse i to retailer j. Thus, it is apparent that amount of flow from i to j must be a decision variable:

$$x_{ij} = \text{Amount of flow from warehouse } i \text{ to retailer } j$$

However, in addition to flows, we have to answer the question whether warehouse i is opened up. If we answer "yes" to this question, it is opened. Let us denote the question with y_i, so that the "yes" answer is $y_i = 1$ and the "no" answer is $y_i = 0$. Clearly, y_i is a decision variable:

$$y_i = \left\{ \begin{array}{ll} 1 & \text{if warehouse } i \text{ opened up} \\ 0 & \text{otherwise} \end{array} \right\}.$$

3.2 Constraints

Demand Constraints: Just as in the transportation problem, demand at each retailer must be satisfied.

$$\sum_{i=1}^{m} x_{ij} \geq d_j \quad j = 1..n$$

Supply Constraints: Supply constraints are a little bit more involved this time. First of all, if warehouse i is not opened up, we can not send any units out of it. In notation, we have:

$$y_i = 0 \implies \sum_{j=1}^{n} x_{ij} = 0$$

On the other hand, if warehouse i is opened, we can send its supply b_i to reatilers:

$$y_i = 1 \implies \sum_{j=1}^{n} x_{ij} \leq b_i$$

When warehouse i is open its supply is b_i, otherwise it is 0. In general, its $y_i b_i$. Thus, we can write the supply constraint as

$$\sum_{j=1}^{n} x_{ij} \leq y_i b_i \quad i = 1..m \; .$$

3.3 Objective Function

Without much discussion, we can directly write down the cost of sending materials from warehouses to retailers:

$$\sum_{i=1}^{j} \sum_{j=1}^{n} c_{ij} x_{ij}$$

In addition to flow costs, we pay F_i by opening warehouse i, and 0 otherwise. In either case, we pay $F_i y_i$. The total cost of warehouses is:

$$\sum_{i=1}^{m} F_i y_I$$

Summing up the flow and warehouse costs, we drive the total cost to be minimized:

$$\text{Minimize } \sum_{i=1}^{j} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} F_i y_I$$

3.4 IP Formulation

Putting all the constraints and the objective function together we obtain the IP formulation:

$$\begin{array}{ll}
\text{Minimize} & \sum_{i=1}^{j} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} F_i y_I \\
\text{Subject to} & \\
& \sum_{i=1}^{m} x_{ij} \geq d_j \quad j = 1..n \\
& \sum_{j=1}^{n} x_{ij} \leq y_i b_i \quad i = 1..m \\
& x_{ij} \geq 0 \quad i = 1..m, \; j = 1..n \\
& y_i \in \{0, 1\} \quad i = 1..m
\end{array}$$

Note that the facility location problem formulation has both binary (y_i) variables and continuous (x_{ij}) variables. When this happens the formulation is called a Mixed Integer Linear Program (MILP).

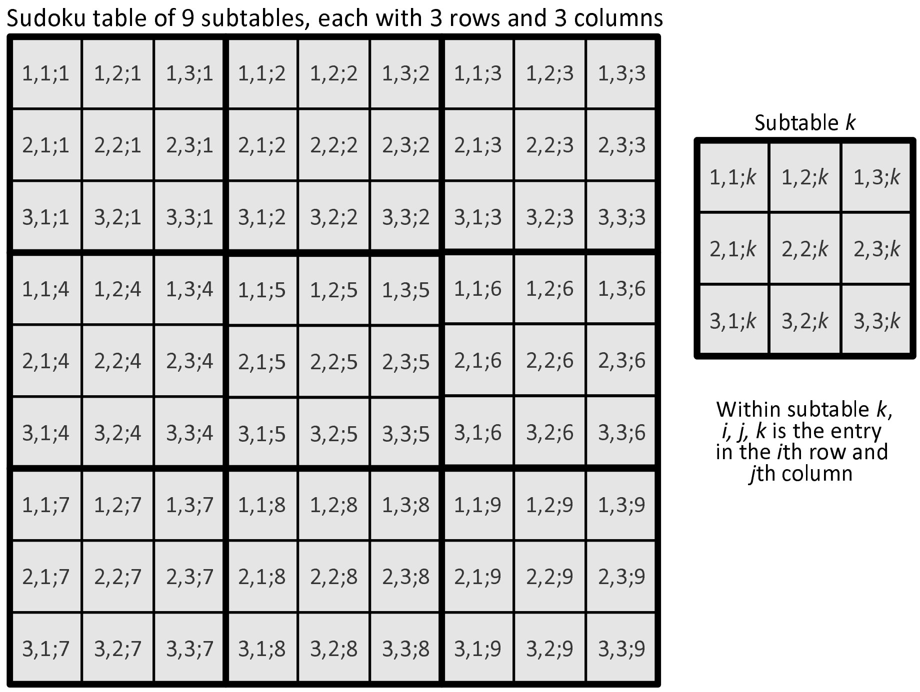


Figure 1: Sudoku table.

4 Sudoku Problem

Sudoku is a number game played on a 9 × 9 table; 9 rows and 9 columns. We consider this table as 9 subtables, each with 3 rows and 3 columns. The cell in row i, column j of subtable k is denoted by $(i, j; k)$. See Figure 1.

In the sudoku game, each cell in the sudoku table must be filled with a number from 1 to 9. But each number can appear exactly once in each row of the sudoku table, once in each column of the sudoku table and once in each subtable. Some numbers already appear in some cells.

4.1 Decision Variables

Let $y^m_{i,j;k}$ be 1 when the cell $(i, j; k)$ contains number m; 0 otherwise.

4.2 Constraints

Supose that some numbers already appear in some cells:

- 3 appears in $(1, 2; 1)$, so $y^3_{1,2;1} = 1$.
- 7 appears in $(2, 1; 1)$, so $y^7_{2,1;1} = 1$.
- 6 appears in $(3, 3; 1)$, so $y^6_{3,3;1} = 1$.
- . . .

- 4 appears in $(1,1;9)$, so $y^4_{1,1;9} = 1$.
- 2 appears in $(2,1;9)$, so $y^2_{2,1;9} = 1$.
- 5 appears in $(3,2;9)$, so $y^5_{3,2;9} = 1$.

Let us make a list of number m already placed in cell $(i,j;k)$ listed above and call it list $\mathcal{L}$. A member of the list is denoted by $[m \text{ @ } (i,j;k)]$. The constraint to specify already given numbers is

$$y^m_{i,j;k} = 1 \text{ for each given number } m \text{ in position } (i,j;k).$$

By using the list, we write these constraints with short-hand notation

$$y^m_{i,j;k} = 1 \text{ for } [m \text{ @ } (i,j;k)] \in \mathcal{L}. \tag{1}$$

Now we write constraints so that each number can appear exactly once in each row, each column and each subtable. The cell indices in the first row of the table is $(1,1;1)$, $(1,2;1)$, $(1,3;1)$, $(1,1;2)$, $(1,2;2)$, $(1,3;2)$, $(1,1;3)$, $(1,2;3)$, $(1,3;3)$. To place number $m = 1$ once in this row, we write

$$y^1_{1,1;1} + y^1_{1,2;1} + y^1_{1,3;1} + y^1_{1,1;2} + y^1_{1,2;2} + y^1_{1,3;2} + y^1_{1,1;3} + y^1_{1,2;3} + y^1_{1,3;3} = 1.$$

To place number $m = 2$ once in this row, we write

$$y^2_{1,1;1} + y^2_{1,2;1} + y^2_{1,3;1} + y^2_{1,1;2} + y^2_{1,2;2} + y^2_{1,3;2} + y^2_{1,1;3} + y^2_{1,2;3} + y^2_{1,3;3} = 1.$$

In general, to place number m once in this row, we write

$$y^m_{1,1;1} + y^m_{1,2;1} + y^m_{1,3;1} + y^m_{1,1;2} + y^m_{1,2;2} + y^m_{1,3;2} + y^m_{1,1;3} + y^m_{1,2;3} + y^m_{1,3;3} = 1.$$

Consider the second row of the table and cells $(2,1;1)$, $(2,2;1)$, $(2,3;1)$, $(2,1;2)$, $(2,2;2)$, $(2,3;2)$, $(2,1;3)$, $(2,2;3)$, $(2,3;3)$ in there. To place number m once in this row, we write

$$y^m_{2,1;1} + y^m_{2,2;1} + y^m_{2,3;1} + y^m_{2,1;2} + y^m_{2,2;2} + y^m_{2,3;2} + y^m_{2,1;3} + y^m_{2,2;3} + y^m_{2,3;3} = 1.$$

Similarly, to place number m once in the third row, we write

$$y^m_{3,1;1} + y^m_{3,2;1} + y^m_{3,3;1} + y^m_{3,1;2} + y^m_{3,2;2} + y^m_{3,3;2} + y^m_{3,1;3} + y^m_{3,2;3} + y^m_{3,3;3} = 1.$$

We can shorten this constraint as

$$\sum_{j=1}^{3} y^m_{3,j;1} + y^m_{3,j;2} + y^m_{3,j;3} = 1 \quad \text{for } m = 1, \ldots 9.$$

Indeed, the constraints for the first 3 rows are

$$\sum_{j=1}^{3} y^m_{i,j;1} + y^m_{i,j;2} + y^m_{i,j;3} = 1 \quad \text{for } i = 1,2,3, m = 1, \ldots 9.$$

These can also be shortened for the first three rows as

$$\sum_{k=1}^{3} \sum_{j=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } i = 1,2,3, m = 1, \ldots 9.$$

For rows 4, 5, 6, we have

$$\sum_{k=4}^{6} \sum_{j=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } i = 1,2,3, m = 1, \ldots 9.$$

For rows 7, 8, 9, we have

$$\sum_{k=7}^{9}\sum_{j=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } i = 1,2,3, m = 1,\ldots 9.$$

In summary, to make sure that each number appears exactly once in each row, we require

$$\sum_{k=1}^{3}\sum_{j=1}^{3} y^m_{i,j;k} = 1; \ \ \sum_{k=4}^{6}\sum_{j=1}^{3} y^m_{i,j;k} = 1; \ \ \sum_{k=7}^{9}\sum_{j=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } i = 1,2,3, m = 1,\ldots 9. \tag{2}$$

We also need constraints to ensure that each number appears only once in each column. For the first column, we require

$$\sum_{k\in\{1,4,7\}}\sum_{i=1}^{3} y^m_{i,1;k} = 1 \quad \text{for } m = 1,\ldots 9.$$

For the second and third colun we need

$$\sum_{k\in\{1,4,7\}}\sum_{i=1}^{3} y^m_{i,2;k} = 1; \quad \sum_{k\in\{1,4,7\}}\sum_{i=1}^{3} y^m_{i,3;k} = 1 \quad \text{for } m = 1,\ldots 9.$$

Then, the constraints for the first three coluns are

$$\sum_{k\in\{1,4,7\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } j = 1,2,3, \ m = 1,\ldots 9.$$

The constraints for columns 4,5,6 are

$$\sum_{k\in\{2,5,8\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } j = 1,2,3, \ m = 1,\ldots 9.$$

The constraints for columns 7,8,9 are

$$\sum_{k\in\{3,6,9\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } j = 1,2,3, \ m = 1,\ldots 9.$$

In summary, to make sure that each number appears exactly once in each column, we require

$$\sum_{k\in\{1,4,7\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k\in\{2,5,8\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k\in\{3,6,9\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } j = 1,2,3, \ m = 1,\ldots 9. \tag{3}$$

We write constraints to make sure that each subtable has only one number m:

$$\sum_{i=1}^{3}\sum_{j=1}^{3} y^m_{i,j;k} = 1 \quad \text{for } k,m = 1,\ldots 9. \tag{4}$$

Finally, we need to ensure that each cell has a number in it:

$$\sum_{m=1}^{9} y^m_{i,j;k} = 1 \quad \text{for } k = 1,\ldots 9, \ i,j = 1,2,3. \tag{5}$$

Now we are ready to put all the constraints in (1-5) together:

$$
\begin{aligned}
y^m_{i,j;k} &= 1 \quad \text{for } [m \ @ \ (i,j;k)] \in \mathcal{L}, \\
\sum_{k=1}^{3}\sum_{j=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k=4}^{6}\sum_{j=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k=7}^{9}\sum_{j=1}^{3} y^m_{i,j;k} &= 1 \quad \text{for } i = 1,2,3, m = 1,\dots 9, \\
\sum_{k\in\{1,4,7\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k\in\{2,5,8\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k\in\{3,6,9\}}\sum_{i=1}^{3} y^m_{i,j;k} &= 1 \quad \text{for } j = 1,2,3,\ m = 1,\dots 9, \\
\sum_{i=1}^{3}\sum_{j=1}^{3} y^m_{i,j;k} &= 1 \quad \text{for } k, m = 1,\dots 9, \\
\sum_{m=1}^{9} y^m_{i,j;k} &= 1 \quad \text{for } k = 1,\dots 9,\ i,j = 1,2,3, \\
y^m_{i,j;k} &\in \{0,1\}.
\end{aligned}
$$

4.3 Objective Function

To finish the formulation we need to write an objective function. In Sudoku game, there is no objective as long as numbers are placed as they should according to constraints in (1-5). Thus we do not prefer any $y^m_{i,j;k} = 1$ to $y^m_{i,j;k} = 0$. Thus, the objective can be maximizing $0y^1_{1,1;1}$ or minimizing $0y^9_{3,3;9}$. They will all give us the objective value of zero, but what we care in Sudoku is only the pacement of the numbers in the table!

4.4 IP Formulation

Max $0y^1_{1,1;1}$

Subject to

$$
\begin{aligned}
y^m_{i,j;k} &= 1 \quad \text{for } [m \ @ \ (i,j;k)] \in \mathcal{L}, \\
\sum_{k=1}^{3}\sum_{j=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k=4}^{6}\sum_{j=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k=7}^{9}\sum_{j=1}^{3} y^m_{i,j;k} &= 1 \quad \text{for } i = 1,2,3, m = 1,\dots 9, \\
\sum_{k\in\{1,4,7\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k\in\{2,5,8\}}\sum_{i=1}^{3} y^m_{i,j;k} = 1; \quad \sum_{k\in\{3,6,9\}}\sum_{i=1}^{3} y^m_{i,j;k} &= 1 \quad \text{for } j = 1,2,3,\ m = 1,\dots 9, \\
\sum_{i=1}^{3}\sum_{j=1}^{3} y^m_{i,j;k} &= 1 \quad \text{for } k, m = 1,\dots 9, \\
\sum_{m=1}^{9} y^m_{i,j;k} &= 1 \quad \text{for } k = 1,\dots 9,\ i,j = 1,2,3, \\
y^m_{i,j;k} &\in \{0,1\}.
\end{aligned}
$$

5 Solved Exercises

1. Merrill Lynch is considering investments into 6 projects: A, B, C, D, E and F. Each project has an initial cost, an expected profit rate (one year from now) expressed as a percentage of the initial cost,

and an associated risk of failure. These numbers are given in the table below:

	A	B	C	D	E	F
Initial cost (in M)	1.3	0.8	0.6	1.8	1.2	2.4
Profit rate	10%	20%	20%	10%	10%	10%
Failure risk	6%	4%	6%	5%	5%	4%

a) Provide a formulation to choose the projects that maximize total expected profit, such that Merrill Lynch does not invest more than 4M dollars and its average failure risk is not over 5%. For example, if Merrill Lynch invests only into A and B, it invests only 2.1M dollars and its average failure risk is (6%+4%)/2=5%.
b) Suppose that if A is chosen, B must be chosen. Modify your formulation.
c) Suppose that if C **and** D are chosen, E must be chosen. Mofdify your formulation.

Solution: a) Let $y_A = 1$ if project A is chosen, $y_A = 0$ otherwise. Define y_B, y_C, y_D, y_E, y_F similarly.

$Max \quad 0.13y_A + 0.16y_B + 0.12y_C + 0.18y_D + 0.12y_E + 0.24y_F$
St
$1.3y_A + 0.8y_B + 0.6y_C + 1.8y_D + 1.2y_E + 2.4y_F \leq 4$
$0.06y_A + 0.04y_B + 0.06y_C + 0.05y_D + 0.05y_E + 0.04y_F \leq 0.05(y_A + y_B + y_C + y_D + y_E + y_F)$
$y_A, y_B, y_C, y_D, y_E, y_F \in \{0, 1\}$.

b) Add : $y_B \geq y_A$.
c) Add : $y_E \geq y_C + y_D - 1$.

6 Exercises

1. After the bad start to the season and unexpected injuries, Dallas Cowboys wants to sign 3 new players. There are five players under consideration. Player 1, 2 and 4 can play the quarter back position and Cowboys want to bring in at least one new player for this position. Each player has a strength measured in s_j and Cowboys want to add at least S units of strength to the team by hiring some of these 5 players. On the other hand, for public relations purposes Cowboys will be careful while signing people who are prone to be convicted for assaults. Player j is likely to have a_j assaults per season and Cowboys imposes a quota of A assaults per season for these players.
a) If player j demands the salary c_j, make up an IP to decide on which players should be signed to minimize the signing budget.
b) If player 1 and 2 are bodies and will come to Dallas only together, add a constraint to a) to guarantee that either they are signed together or they are not signed at all.
c) If player 1 hates player 3 and will not come if player 3 comes, add an appropriate constraint that does not allow signing player 1 and 3 together.

2. Suppose that you will play a word-construction game. You are assigned n_a, n_b, $\ldots$, n_z copies of letters a, b, $\ldots$, z respectively. That is, if $n_b = 8$ then you can use the letter b at most 8 times in your words. These words must be in an English dictionary, say we use D to denote all the words in the dictionary. Then, *abacus* $\in D$, *abode* $\in D$, *zinc* $\in D$, basically D is the set of all English words. Suppose that for every word you construct, you obtain points equivalent to the length of that word. For example, if you are assigned $n_a = 1$, $n_e = 2$, $n_i = 5$, $n_m = 4$, $n_n = 1$, $n_x = 1$, $n_z = 2$ $(a,e,e,i,i,i,i,i,m,m,m,m,n,x,z,z)$ and no other letter, you should construct words *minimize* and *maximize* scoring 16 points instead of *zen* and *maze* scoring 7 points. For an assignment of letters (i.e., n_a, n_b, $\ldots$, n_z given and known), provide an IP that will choose words so that your score is

maximized. Hint: Let $x_i = 1$ if the ith word in D is constructed, 0 otherwise. Let a_i be the number of letter "a"'s in the ith word, similarly define b_i to z_i for all $i \in D$.

3. Consider the Market Clearing model presented in the Formulation Chapter. Suppose that there is a fixed cost K_{ij} of using each transportation channel from supplier i to buyer j. This cost is independent of the units that are transported form i to j, it is incurred if there is some, no matter how small or large, shipment. In addition, we want to enforce that each product k is supplied by at most one supplier. Note that this restriction will make the market for each product a monopoly. Provide a formulation to market clearing problem with these additions.

4. Suppose that you are moving to a new apartment and you have only two boxes to pack your stuff in. The first box carries 22 kg and the second carries 28 kg. The weight and the value of your belongings are:

Item	A	B	C	D	E	F	G	H
Weight (kg)	10	9	15	3	11	6	3	4
Value (in \$100)	5	2	7	6	1	6	8	6

a) Provide a formulation to maximize the value of items you can carry with these two boxes.

b) Since the box capacity is limited, you cannot fit all your items into boxes. Can you identify an item, which will always (in all optimal solutions) be put in one of the boxes? Justify your answer.

5. Consider a smaller Sudoku game played on 4 subtables each with 2 rows and 2 columns. Following numbers are already placed:

- 2 appears in $(1, 2; 1)$
- 3 appears in $(2, 1; 1)$
- 3 appears in $(1, 2; 2)$
- 1 appears in $(1, 1; 3)$
- 3 appears in $(2, 2; 3)$
- 3 appears in $(1, 1; 4)$
- 1 appears in $(2, 2; 4)$

We are to place numbers from 1 to 4 into the remaining cells in this smaller Sudoku table. Provide an explicit IP formulation (without using sum notation) to solve this Sudoku problem.